Terms used **data** and **forward** and **processing element** and **update** and **register** and **address** and **memory address** and **flag** and **network**

Found **42,831** of **148,162**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Exploiting task-level concurrency in a programmable network interface

Hyong-youb Kim, Vijay S. Pai, Scott Rixner

June 2003 **ACM SIGPLAN Notices , Proceedings of the ninth ACM SIGPLAN symposium on Principles and practice of parallel programming**, Volume 38 Issue 10

Full text available: pdf(191.35 KB) Additional Information: full citation, abstract, references, citings, index terms

Programmable network interfaces provide the potential to extend the functionality of network services but lead to instruction processing overheads when compared to application-specific network interfaces. This paper aims to offset those performance disadvantages by exploiting task-level concurrency in the workload to parallelize the network interface firmware for a programmable controller with two processors. By carefully partitioning the handler procedures that process various events related to ...

Keywords: ethernet, firmware, parallel programming, programmable network interface

2 Maps: a compiler-managed memory system for raw machines

Rajeev Barua, Walter Lee, Saman Amarasinghe, Anant Agarwal

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture**, Volume 27 Issue 2

Full text available: pdf(231.60 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper describes Maps, a compiler managed memory system for Raw architectures. Traditional processors for sequential programs maintain the abstraction of a unified memory by using a single centralized memory system. This implementation leads to the infamous "Von Neumann bottleneck," with machine performance limited by the large memory latency and limited memory bandwidth. A Raw architecture addresses this problem by taking advantage of the rapidly increasing transistor budget to move much of ...

3 The EM-X parallel computer: architecture and basic performance

Yuetsu Kodama, Hirohumi Sakane, Mitsuhisa Sato, Hayato Yamana, Shuichi Sakai, Yoshinori Yamaguchi

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture**, Volume 23 Issue 2

Full text available: pdf(1.04 MB) Additional Information: full citation, abstract, references, citings, index terms

Latency tolerance is essential in achieving high performance on parallel computers for remote function calls and fine-grained remote memory accesses. EM-X supports

h c g e cf c

interprocessor communication on an execution pipeline with small and simple packets. It can create a packet in one cycle, and receive a packet from the network in the on-chip buffer without interruption. EM-X invokes threads on packet arrival, minimizing the overhead of thread switching. It can tolerate communication latency by using ...

4 A survey of commercial parallel processors

Edward Gehringer, Janne Abullarade, Michael H. Gulyn

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

Full text available: pdf(2.96 MB) Additional Information: full citation, abstract, citings, index terms

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel iPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication. Also, the share ...

5 The NYU Ultracomputer—designing a MIMD, shared-memory parallel machine (Extended Abstract)

Allan Gottlieb, Ralph Grishman, Clyde P. Kruskal, Kevin P. McAuliffe, Larry Rudolph, Marc Snir

April 1982 **Proceedings of the 9th annual symposium on Computer Architecture**

Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

We present the design for the NYU Ultracomputer, a shared-memory MIMD parallel machine composed of thousands of autonomous processing elements. This machine uses an enhanced message switching network with the geometry of an Omega-network to approximate the ideal behavior of Schwartz's paracomputer model of computation and to implement efficiently the important fetch-and-add synchronization primitive. We outline the hardware that would be required to build a 4096 processor system using 1990' ...

6 The NYU ultracomputer—designing a MIMD, shared-memory parallel machine

Allan Gottlieb, Ralph Grishman, Clyde P. Kruskal, Kevin P. McAuliffe, Larry Rudolph, Marc Snir

August 1998 **25 years of the international symposia on Computer architecture (selected papers)**

Full text available: pdf(1.74 MB) Additional Information: full citation, references, index terms

7 Embedded applications: AES and the cryptonite crypto processor

Dino Oliva, Rainer Buchty, Nevin Heintze

October 2003 **Proceedings of the 2003 international conference on Compilers, architectures and synthesis for embedded systems**

Full text available: pdf(346.09 KB) Additional Information: full citation, abstract, references, index terms

CRYPTONITE is a programmable processor tailored to the needs of crypto algorithms. The design of CRYPTONITE was based on an in-depth application analysis in which standard crypto algorithms (AES, DES, MD5, SHA-1, etc) were distilled down to their core functionality. We describe this methodology and use AES as a central example. Starting with a functional description of AES, we give a high level account of how to implement AES efficiently in hardware, and present several novel optimizations (whic ...

Keywords: AES, architecture, cryptography, high-bandwidth, high-speed, processor, round key generation, software implementation

8 A two-tier memory architecture for high-performance multiprocessor systems

T. M. Nguyen, V. P. Srini, A. M. Despain

June 1988 **Proceedings of the 2nd international conference on Supercomputing**

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

Performance of high-speed multiprocessor systems is limited by the available bandwidth to memory and the need to synchronize write sharable data. This paper presents a new memory system that separates synchronization related data from others. The memory system has two tiers: synchronization memory and high bandwidth (HB) memory. The synchronization memory consists of snooping caches connected to a bus and is used to store synchronization variables such as locks and semaphores. The H ...

9 MULTILISP: a language for concurrent symbolic computation

Robert H. Halstead

October 1985 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 7 Issue 4

Full text available: pdf(3.30 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Multilisp is a version of the Lisp dialect Scheme extended with constructs for parallel execution. Like Scheme, Multilisp is oriented toward symbolic computation. Unlike some parallel programming languages, Multilisp incorporates constructs for causing side effects and for explicitly introducing parallelism. The potential complexity of dealing with side effects in a parallel context is mitigated by the nature of the parallelism constructs and by support for abstract data types: a recommende ...

10 Data-Driven and Demand-Driven Computer Architecture

Philip C. Treleaven, David R. Brownbridge, Richard P. Hopkins

January 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 1

Full text available: pdf(4.14 MB) Additional Information: full citation, references, citings, index terms

11 Balancing performance and flexibility with hardware support for network architectures

Ilija Hadžić, Jonathan M. Smith

November 2003 **ACM Transactions on Computer Systems (TOCS)**, Volume 21 Issue 4

Full text available: pdf(719.03 KB) Additional Information: full citation, abstract, references, index terms

The goals of performance and flexibility are often at odds in the design of network systems. The tension is common enough to justify an architectural solution, rather than a set of context-specific solutions. The Programmable Protocol Processing Pipeline (P4) design uses programmable hardware to selectively accelerate protocol processing functions. A set of field-programmable gate arrays (FPGAs) and an associated library of network processing modules implemented in hardware are augmented with so ...

Keywords: FPGA, P4, computer networking, flexibility, hardware, performance, programmable logic devices, programmable networks, protocol processing

12 Resource allocation in a high clock rate microprocessor

Michael Upton, Thomas Huff, Trevor Mudge, Richard Brown

November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 , 28 Issue 11 , 5

Full text available: pdf(1.10 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper discusses the design of a high clock rate (300MHz) processor. The architecture is described, and the goals for the design are explained. The performance of three processor models is evaluated using trace-driven simulation. A cost model is used to estimate the resources required to build processors with varying sizes of on-chip memories, in both single and dual issue models. Recommendations are then made to increase the effectiveness of each of the models.

Keywords: decoupled architecture, floating point latencies, nonblocking cache, pipelining, prefetching, resource allocation, superscalar

13 EPIC compilation: Speculative register promotion using Advanced Load Address Table (ALAT)

Jin Lin, Tong Chen, Wei-Chung Hsu, Pen-Chung Yew

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available: pdf(891.95 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The pervasive use of pointers with complicated patterns in C programs often constrains compiler alias analysis to yield conservative register allocation and promotion. Speculative register promotion with hardware support has the potential to more aggressively promote memory references into registers in the presence of aliases. This paper studies the use of the Advanced Load Address Table (ALAT), a data speculation feature defined in the IA-64 architecture, for speculative register promotion. An ...

14 Fine-grained mobility in the Emerald system

Eric Jul, Henry Levy, Norman Hutchinson, Andrew Black

February 1988 **ACM Transactions on Computer Systems (TOCS)**, Volume 6 Issue 1

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Emerald is an object-based language and system designed for the construction of distributed programs. An explicit goal of Emerald is support for object mobility; objects in Emerald can freely move within the system to take advantage of distribution and dynamically changing environments. We say that Emerald has fine-grained mobility because Emerald objects can be small data objects as well as process objects. Fine-grained mobility allows us to apply mobility in new ways but presents implemen ...

15 Specialized merge processor networks for combining sorted lists

Lee A. Hollaar

September 1978 **ACM Transactions on Database Systems (TODS)**, Volume 3 Issue 3

Full text available: pdf(985.06 KB) Additional Information: full citation, abstract, references, citings, index terms

In inverted file database systems, index lists consisting of pointers to items within the database are combined to form a list of items which potentially satisfy a user's query. This list merging is similar to the common data processing operation of combining two or more sorted input files to form a sorted output file, and generally represents a large percentage of the computer time used by the retrieval system. Unfortunately, a general purpose digital computer is better suited for complica ...

Keywords: backend processors, binary tree networks, computer system architecture, full text retrieval systems, inverted file databases, nonnumeric processing, pipelined networks, sorted list merging

16 A microprocessor-controlled asynchronous circuit switching network

Tse-yun Feng, Chuan-lin Wu, Dharma P. Agrawal

April 1979 **Proceedings of the 6th annual symposium on Computer architecture**

Full text available: pdf(839.81 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes an asynchronous circuit switching network for multiple-processor systems. Several circuit switching networks for various applications have been proposed and constructed. However, there are problems associated with these networks. The asynchronous circuit switching network possesses several features that can solve these

problems. A three-stage fully connected topology is utilized to construct the network. Each switching element is functionally and physically identical an ...

17 Speculative synchronization: applying thread-level speculation to explicitly parallel applications

José F. Martínez, Josep Torrellas

October 2002 **Proceedings of the 10th international conference on Architectural support for programming languages and operating systems**, Volume 36 , 30 , 37 Issue 5 , 5 , 10

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings

Barriers, locks, and flags are synchronizing operations widely used programmers and parallelizing compilers to produce race-free parallel programs. Often times, these operations are placed suboptimally, either because of conservative assumptions about the program, or merely for code simplicity.We propose *Speculative Synchronization,* which applies the philosophy behind Thread-Level Speculation (TLS) to explicitly parallel applications. Speculative threads execute past active barriers, busy ...

18 A preliminary architecture for a basic data-flow processor

Jack B. Dennis, David P. Misunas

December 1974 **ACM SIGARCH Computer Architecture News , Proceedings of the 2nd annual symposium on Computer architecture**, Volume 3 Issue 4

Full text available: pdf(675.64 KB) Additional Information: full citation, abstract, references, citings

A processor is described which can achieve highly parallel execution of programs represented in data-flow form. The language implemented incorporates conditional and iteration mechanisms, and the processor is a step toward a practical data-flow processor for a Fortran-level data-flow language. The processor has a unique architecture which avoids the problems of processor switching and memory/processor interconnecion that usually limit the degree of realizable concurrent processing. The architect ...

19 The KScalar simulator

J. C. Moure, Dolores I. Rexachs, Emilio Luque

March 2002 **Journal on Educational Resources in Computing (JERIC)**, Volume 2 Issue 1

Full text available: pdf(493.35 KB) Additional Information: full citation, abstract, references, index terms

Modern processors increase their performance with complex microarchitectural mechanisms, which makes them more and more difficult to understand and evaluate. KScalar is a graphical simulation tool that facilitates the study of such processors. It allows students to analyze the performance behavior of a wide range of processor microarchitectures: from a very simple in-order, scalar pipeline, to a detailed out-of-order, superscalar pipeline with non-blocking caches, speculative execution, and comp ...

Keywords: Education, pipelined processor simulator

20 System architectures for computer music

John W. Gordon

June 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 2

Full text available: pdf(4.61 MB) Additional Information: full citation, abstract, references, index terms, review

Computer music is a relatively new field. While a large proportion of the public is aware of computer music in one form or another, there seems to be a need for a better understanding of its capabilities and limitations in terms of synthesis, performance, and recording hardware. This article addresses that need by surveying and discussing the architecture of existing computer music systems. System requirements vary according to what the system will be used for. Common uses for co ...

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	IS&R	L1	1153	(712/235,22,28,32,35).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:42	
2	IS&R	L2	459	(345/506,505).CCLS.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:42	
3	IS&R	L3	0	("0001or2").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:43	
4	BRS	L4	1600	1 or 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:43	
5	BRS	L5	203	4 and shared adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:43	
6	BRS	L6	39	5 and processing adj element	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:44	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
7	BRS	L7	26	6 and address near3 register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:47	
8	BRS	L8	96	7 and time-to-live or hop-count	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:46	
9	BRS	L9	0	7 and (time-to-live or hop-count)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:46	
10	BRS	L10	0	7 and time-to-live	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:46	
11	BRS	L11	0	7 and hop-count	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:46	
12	BRS	L12	896	time-to-live or hop-count	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:47	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
13	BRS	L13	0	7 and 12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:47	
14	BRS	L14	25	7 and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:48	
15	BRS	L15	21	14 and update$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:48	
16	BRS	L16	14	15 and transmit$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2005/01/09 09:49	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
1	BRS	L1	1208	data near3 forward$5 and processing adj element$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:38	
2	BRS	L2	183	L1 and address adj register and (CAM or queue or FIFO)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:38	
3	BRS	L3	49	L2 and data adj field and address adj field	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:38	
4	BRS	L4	13	L3 and memory adj address	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:38	
5	BRS	L7	80187	"13" and telecommunication	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:48	
6	BRS	L6	7	L5 and switch	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:38	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
7	BRS	L8	96	data near3 forward$5 same processing adj element	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:44	
8	BRS	L10	0	9 and data adj field and address adj field	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:45	
9	BRS	L11	0	5 and telecommunication	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 07:47	
10	BRS	L12	7	5 and shared adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 08:01	
11	BRS	L13	7	12 and compar$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 08:02	
12	BRS	L14	7	12 and compar$5 near6 address	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 08:03	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
13	BRS	L15	7	14 and network	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 08:06	
14	BRS	L5	7	L4 and update$5 near3 data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:26	
15	BRS	L16	0	4 and update$5 near3 data and 'time-to-live'	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:27	
16	BRS	L17	812	'time-to-live'	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:31	
17	BRS	L18	5	1 and 17	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:27	
18	BRS	L9	5	8 and address adj register and (CAM or queue or FIFO)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:30	

	Type	L #	Hits	Search Text	DBs	Time Stamp	Comments
19	BRS	L19	0	4 and 'time-to-live'	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	2004/12/29 09:31	